CONCURRENT ENGINEERING: Research and Applications

Integrated Clustering Approach to Developing Technology for Functional Feature and Engineering Specification-based Reference Design Retrieval

Yuh-Jen Chen,[1,*] Yuh-Min Chen,[1] Hui-Chuan Chu,[2] Chin-Bin Wang,[3] Der-Chian Tsaih[3] and Hui-Mei Yang[4]

[1] *Institute of Manufacturing Engineering, National Cheng Kung University, Tainan, Taiwan, ROC*
[2] *National University of Taiwan, Tainan, Taiwan, ROC*
[3] *Department of Information Management, Nan-Hua University, Chia-Yi, Taiwan, ROC*
[4] *Department of Business Administration, Tatung Institute of Commerce and Technology, Chia-Yi, Taiwan, ROC*

Abstract: Engineering design is a complex activity, and is heavily reliant on the know-how of engineering designers. Hence, capturing, storing, and reusing design information, design intent, and underlining design knowledge to support design activities is a key issue in engineering knowledge management.

To meet the demand for engineering designers regarding functional feature and engineering specification-based knowledge resources, this study proposes a novel scheme for functional feature and engineering specification-based reference design retrieval using an integrated clustering approach for providing engineering designers with easy access to relevant reference design and associated knowledge. The research objectives can be achieved by performing the following five tasks: (i) designing a functional feature and engineering specification-based reference design retrieval process, (ii) developing a functional feature and engineering specification representation, (iii) investigating and integrating ART1 (adaptive resonance theory 1) neural network, GA (genetic algorithm), and fuzzy ART (fuzzy adaptive resonance theory) clustering techniques, and (iv) implementing a functional feature and engineering specification-based reference design retrieval mechanism and experimenting with an example. The retrieval process involves three steps: functional feature and engineering specification-based query, similar design case search and retrieval, and similar design case ranking. The techniques involved include: (i) a binary code-based representation for functional feature and an EXPRESS language-based representation for engineering specification, (ii) ART1 neural network and genetic algorithm for functional feature-based similar design case clustering, (iii) fuzzy ART for engineering specification-based similar design case clustering, (iv) similarity calculation for ranking similar design cases, and (v) a case-based representation for designed entities.

Key Words: engineering knowledge management, reference design retrieval, adaptive resonance theory (ART), genetic algorithm (GA), fuzzy adaptive resonance theory (fuzzy ART).

1. Introduction

Knowledge is an important asset for any enterprise owing to global competition and the rapid development of information technology in the 21st knowledge economy era. Knowledge Management (KM) is considered an important factor in improving the competitive edge of enterprises. Consequently, effectively capturing, storing, and re-using useful knowledge within an organization to accumulate intellectual capital is essential for modern businesses.

Engineering design [4,5,15] is the systematic process of identifying 'customer requirements', translating them into the 'functional requirements' of a product, and then mapping these functional requirements into 'functional features and engineering specifications' that can be economically met during manufacture, by exploiting creativity, scientific principles, and technical knowledge. This design procedure can be considered a process of product design information or a process of transforming product design information. Various design states contain different extents of product design information; however, the transformation from one product design information state to another results from a decision process, driven by the design knowledge and experience of engineering designers. Effectively organizing, storing, and retrieving such knowledge and experience is a key factor in increasing product development capability and quality, and reducing the development cycle time and cost. Consequently, organizing, storing, and retrieving product design information, design intent, and underlying design knowledge constitute the foundation of engineering knowledge management.

Owing to their high dependency on information, many companies can significantly improve the performance and efficiency of their service or product delivery

*Author to whom correspondence should be addressed.
E-mail: denis@cacerl.ime.ncku.edu.tw
Figures 3 and 9–12 appear in color online: http://cer.sagepub.com

1063-293X/05/04 0257–20 $10.00/0 DOI: 10.1177/1063293X05059805

by adopting traditional information systems or knowledge management systems. These systems generally comprise a set of interrelated computer-based elements that retrieve, process, store, and distribute information to support activities at either the enterprise or inter-enterprise levels [3,7,8]. No technology has yet been developed for retrieving engineering knowledge (such as design intent and experience) by querying functional features and its engineering specifications. Such retrieval creates a bottleneck in sharing valuable product information and engineering knowledge, and thus a satisfactory engineering knowledge management system has not yet been realized.

This study proposes an integrated clustering approach to developing technology for functional feature and engineering specification-based reference design retrieval to provide engineering designers with easy access to relevant reference designs and associated knowledge. It is part of the findings of our research on collaborative knowledge management in allied concurrent engineering. This study also presents a distributed knowledge management framework and an engineering knowledge management cycle alongside the development of the functional feature and engineering specification-based reference design retrieval technology.

Development of the research includes the following tasks: (i) designing a functional feature and engineering design-based reference design retrieval process, (ii) developing a functional feature and engineering specification representation, (iii) investigating and integrating ART1 (adaptive resonance theory 1) neural network, GA (genetic algorithm), and fuzzy ART (fuzzy adaptive resonance theory) clustering techniques, (iv) implementing a functional feature and engineering specification-based reference design retrieval mechanism and experimenting with an example. Technology related techniques comprise a binary code-based representation for functional feature and an EXPRESS language-based representation for engineering specification, ART1 neural network and genetic algorithm for functional feature-based clustering of similar design cases, fuzzy ART for engineering specification-based clustering of similar design cases, similarity calculation for ranking of similar design cases, and a case-based representation for designed entities.

2. Research Scope

This section first introduces the framework of collaborative knowledge management in allied concurrent engineering [12]. The functional framework of engineering knowledge management is then presented based on the concept of knowledge management for supporting the collaborative knowledge management framework.

2.1 Collaborative Engineering Knowledge Management

Two types of knowledge management are proposed to support levels of knowledge management in an allied concurrent engineering project, namely personal knowledge management and team knowledge management. Personal knowledge management involves knowledge management of individual team members, and connects a team knowledge management unit, while team knowledge management involves knowledge management for a team of project members. It may be equipped with a knowledge repository and be able to communicate with other team knowledge management units.

Furthermore, two levels of knowledge repositories are designed for knowledge storage, namely personal knowledge repository and team knowledge repository. A personal knowledge repository, which is managed by personal knowledge management, is a private storage area for individual team members. Meanwhile, a team knowledge repository comprises a group storage area managed by team knowledge management.

Based on the characteristics of allied concurrent engineering [12,16], an allied concurrent engineering project may consist of several collaborative processes, involving several individual and/or collaborative activities. Therefore, to support knowledge management, a team knowledge management unit may play the role as project knowledge management, process knowledge management, or collaborative activity knowledge management, forming a hierarchical, distributed, flexible, and dynamic-configurable knowledge management framework for allied concurrent engineering, as shown in Figure 1.

2.2 Functional Framework of Engineering Knowledge Management

Based on the collaborative engineering knowledge management framework above, the functional framework for a collaborative engineering knowledge management system is designed to support knowledge-intensive activities in engineering design. Figure 2 shows the functional framework as a knowledge management life cycle, which consists of the creation, capture, compilation and storage, and retrieval/reuse/query of engineering knowledge. The elements of the knowledge management life cycle are briefly outlined below.

2.2.1 *ENGINEERING KNOWLEDGE CREATION*

Engineering designers generally use different methods of structured design to perform and achieve their design objectives. Examination of engineering designer behavior reveals four commonly employed structured design methods: feature-based design, engineering change, design by modification and design by reference.

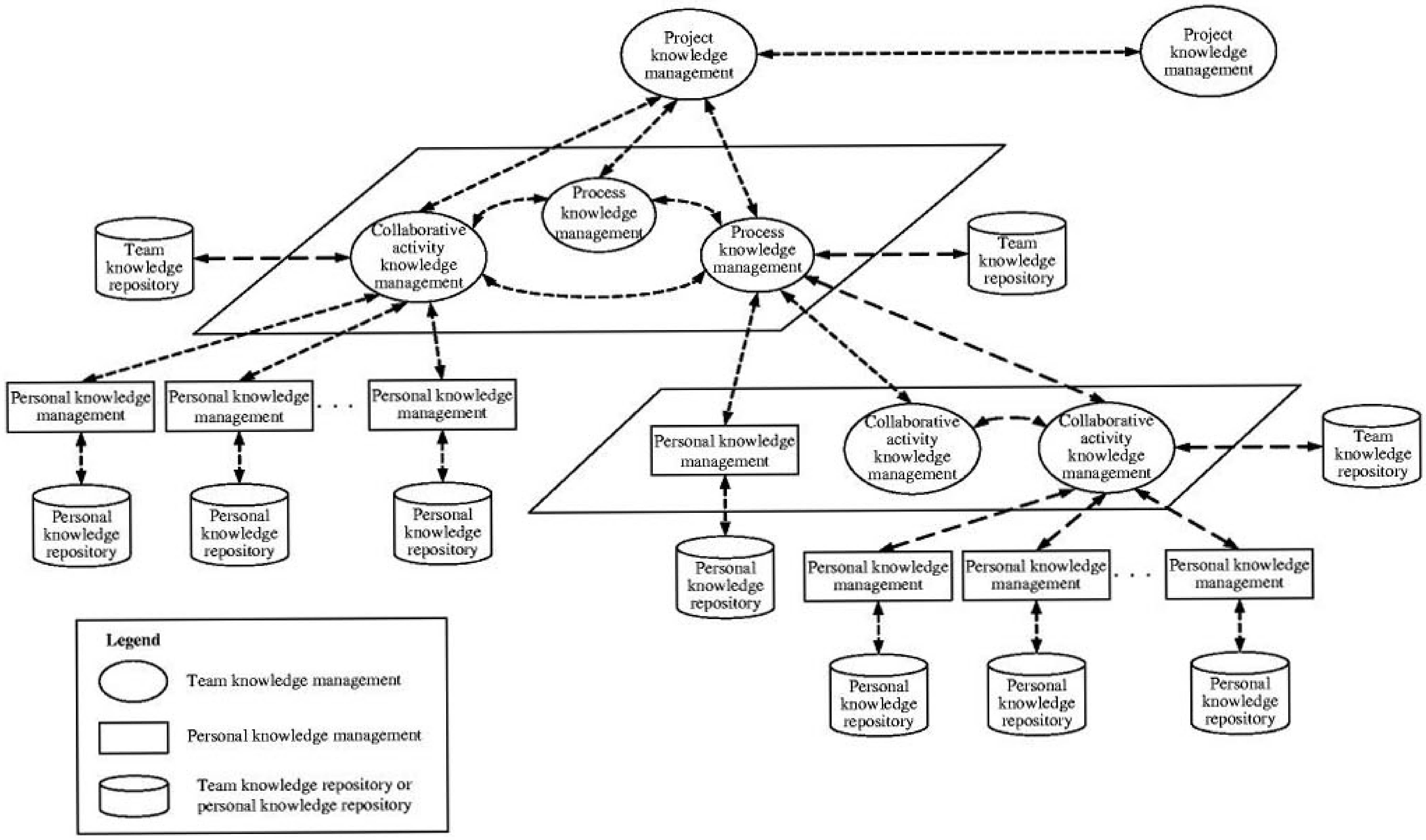


Figure 1. Collaborative knowledge management framework.

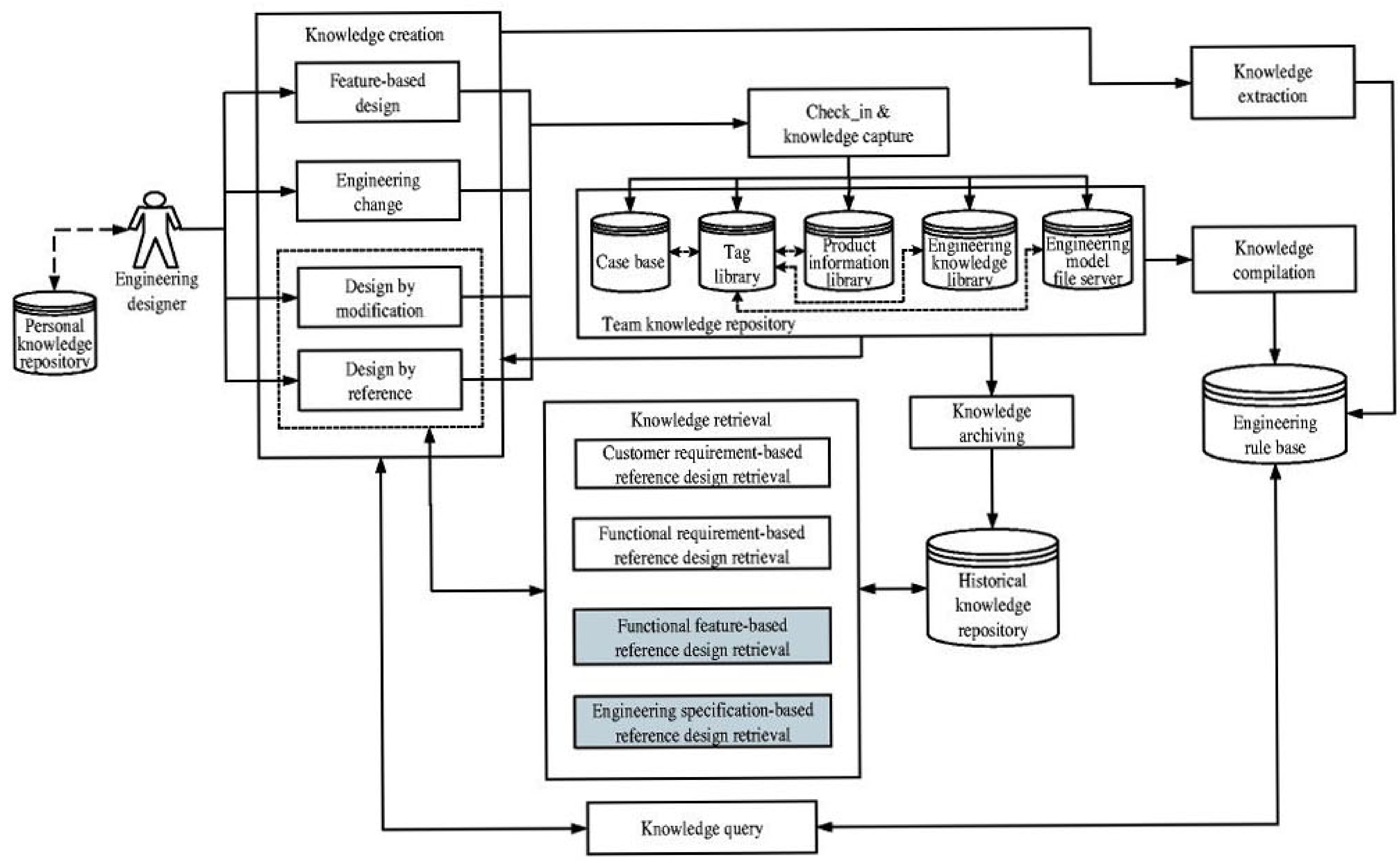


Figure 2. Functional Framework of Engineering Knowledge Management.

In feature-based design, product modeling uses a library of 2D or 3D features as design primitives. Product functional requirements are transformed into functional features, which are then converted into design specifications and manufacturing features. The engineering knowledge involved in feature-based design includes design intent, engineering principles, design experience, creativity, and product information. Moreover, product information can be subdivided into the areas of customer needs, functional requirements, functional features, and engineering specifications. Engineering change is usually defined as a change in the form, fit, or function of a product or part to satisfy customer requirements. Engineering change is triggered by an engineering change request, following which the engineering change is proposed, investigated, authorized/rejected, executed, reviewed, and archived in an orderly, structured design manner. Knowledge of engineering change can be specialized as change knowledge, which can be categorized into a reason for change and the content of the change, as well as the applied engineering principles. Design by modification/design by reference is employed to reduce the design time and increase the working efficiency of engineering designers. This method enables a most similar engineering model to be retrieved from the historical knowledge repository according to the product information, which is then slightly modified to generate a new engineering model, or provides a reference model for a new design. Engineering knowledge is involved in both design by modification and design by reference, and includes product information, design intent, engineering principles and design experience.

2.2.2 *ENGINEERING KNOWLEDGE EXTRACTION, COMPILATION, AND STORAGE*

Design intent, applied engineering principles and heuristics, and information related to engineering collaboration can be extracted during the engineering design, and are associated with the design object as notes for reference. Once an engineering model is completed and checked into a project knowledge repository, product information and knowledge relating to the engineering model are captured and stored in the product information and engineering knowledge libraries, respectively. Additionally, the product information and engineering knowledge are compiled in rule format and deposited in an engineering rule base. Upon completion of a design project, the engineering models and associated knowledge are stored in a historical knowledge repository for future reuse.

2.2.3 *ENGINEERING KNOWLEDGE RETRIEVAL/REUSE/QUERY*

Product information and engineering knowledge can be retrieved when an engineering model is examined or copied from the project knowledge repository. Similarly, historical engineering models, related product information and engineering knowledge can also be referenced or copied to provide a reference for new projects. Moreover, engineering designers can conveniently query engineering knowledge by using the knowledge query function to solve related design problems.

Some studies [17–19] have examined enabling technology in the proposed functional framework of engineering knowledge management, including 'capturing,' 'representation,' 'storage,' and 'retrieval' of engineering knowledge. Meanwhile, the 'retrieval' portion can be classified into customer requirement-based reference design retrieval, functional requirement-based reference design retrieval, and functional feature and engineering specification-based reference design retrieval, to satisfy the knowledge demand of engineering designers regarding the phases of 'customer requirement establishment,' 'functional requirement establishment,' and 'functional feature and engineering specification design,' respectively.

To pursue the proposed framework and system more completely, this study focuses primarily on functional feature and engineering specification-based reference design retrieval, as indicated by the shaded area surrounded by the broken line in Figure 2.

3. Functional Feature and Engineering Specification-based Reference Design Retrieval

This section first describes the process of functional feature and engineering specification-based reference design retrieval. Subsequently, a number of crucial techniques used in the functional feature and engineering specification-based reference design retrieval process are presented, including: (i) a binary code-based representation for functional feature and an EXPRESS language-based representation for engineering specification, (ii) an integrated clustering approach containing the ART1 neural network and GA for functional feature-based similar design case clustering, and fuzzy ART for engineering specification-based similar design case clustering, (iii) similarity calculation for similar design case ranking, and (iv) case-based representation for designed entities. These techniques help to realize a functional feature and engineering specification-based reference design retrieval mechanism.

3.1 Functional Feature and Engineering Specification-based Reference Design Retrieval Process

Functional feature and engineering specification-based reference design retrieval attempts to retrieve the most similar design cases from the reference design case

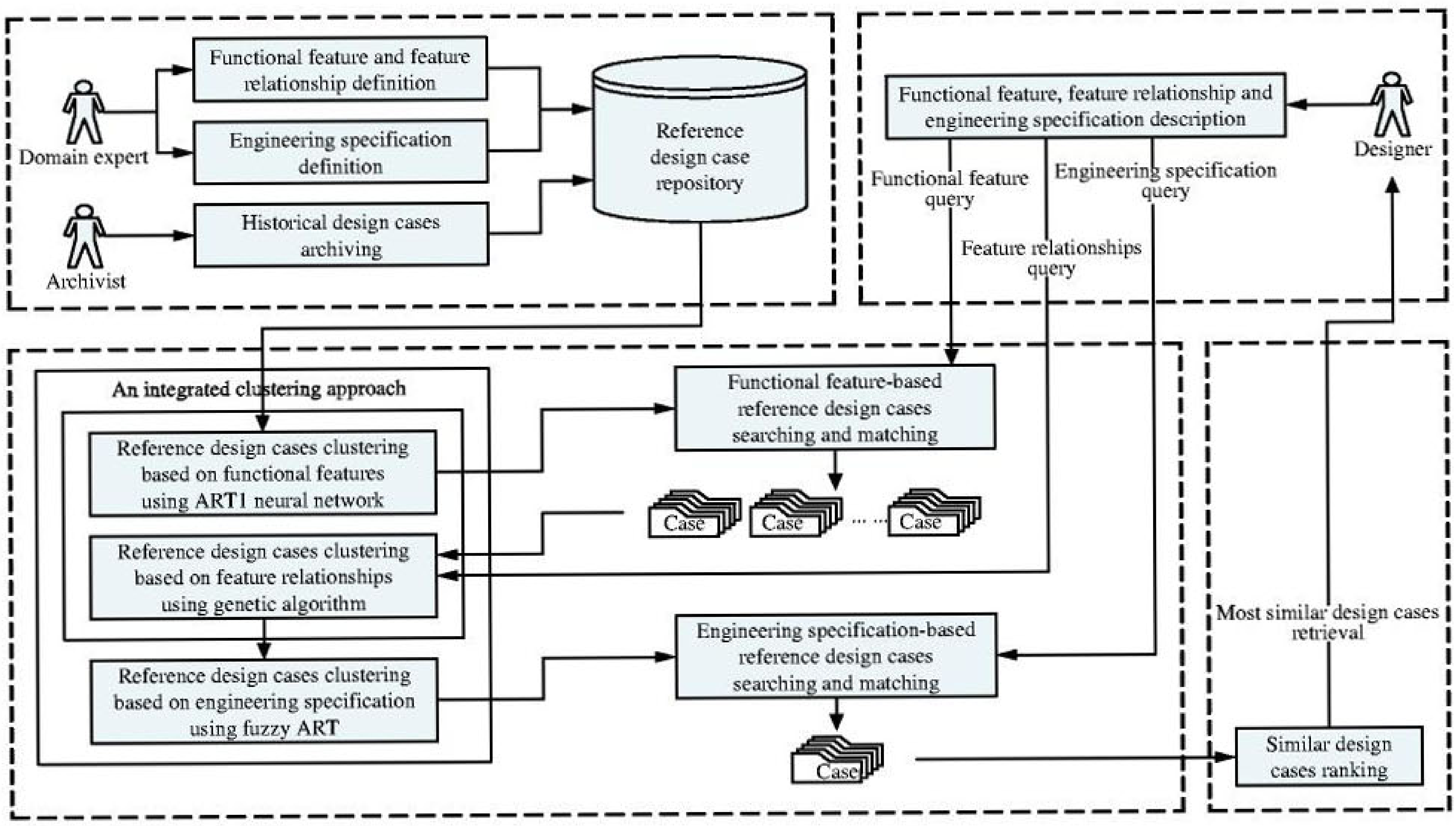


Figure 3. Functional feature and engineering specification-based reference design retrieval process.

repository as references based on user functional feature and engineering specification-based query. To achieve the above goal, the functional feature and engineering specification-based reference design retrieval process is designed based on the concept of the case-based reasoning (CBR) cycle, as illustrated in Figure 3.

Four main steps are involved in running functional feature and engineering specification-based reference design retrieval process.

Step 1: Definition and representation of functional feature, feature relationship, and engineering specification for historical design cases. Functional feature, feature relationship, and engineering specification for historical design cases are defined and represented by experts in engineering design according to the investigation of feature-based design and product data representation. After that, all completed design cases can be represented by the defined functional features and engineering specifications, providing the basis for searching and matching functional feature and engineering specification-based reference design cases.

Step 2: Establishment of functional feature and engineering specification-based query. Using the defined and represented functional features and engineering specifications, designers can easily establish a functional feature (including feature relationship) and engineering specification query as a search target to trigger the execution of functional feature and engineering specification-based reference design retrieval process.

Step 3: Functional feature and engineering specification-based reference design case searching and matching. Similar reference design case searching and matching is conducted based on *Step 1* and *Step 2* above. In this section, this study proposes an integrated clustering approach, which combines the three techniques of ART1 neural network, GA and fuzzy ART. The first two techniques are applied first, and reference design cases are selected in terms of functional features and feature relationships, while the latter case is mainly used to refine the retrieved reference design cases in terms of engineering specifications.

Step 4: Similar design cases ranking. Before the refined reference design cases are delivered to the designer, they should be ranked in the order of calculated coefficients of similarity. The most similar design case is then presented to the designer as a possible reference or scenario to be reused for the design problem under consideration.

3.2 Definition and Representation of Functional Feature and Engineering Specification

This subsection employs object-oriented modeling techniques [1] and data abstraction to model the feature-based part definition data. A part and its elements are represented in terms of objects.

Figure 4 shows that a part is the aggregation of functional features, feature relationships, and engineering specifications. Each feature relationship is associated

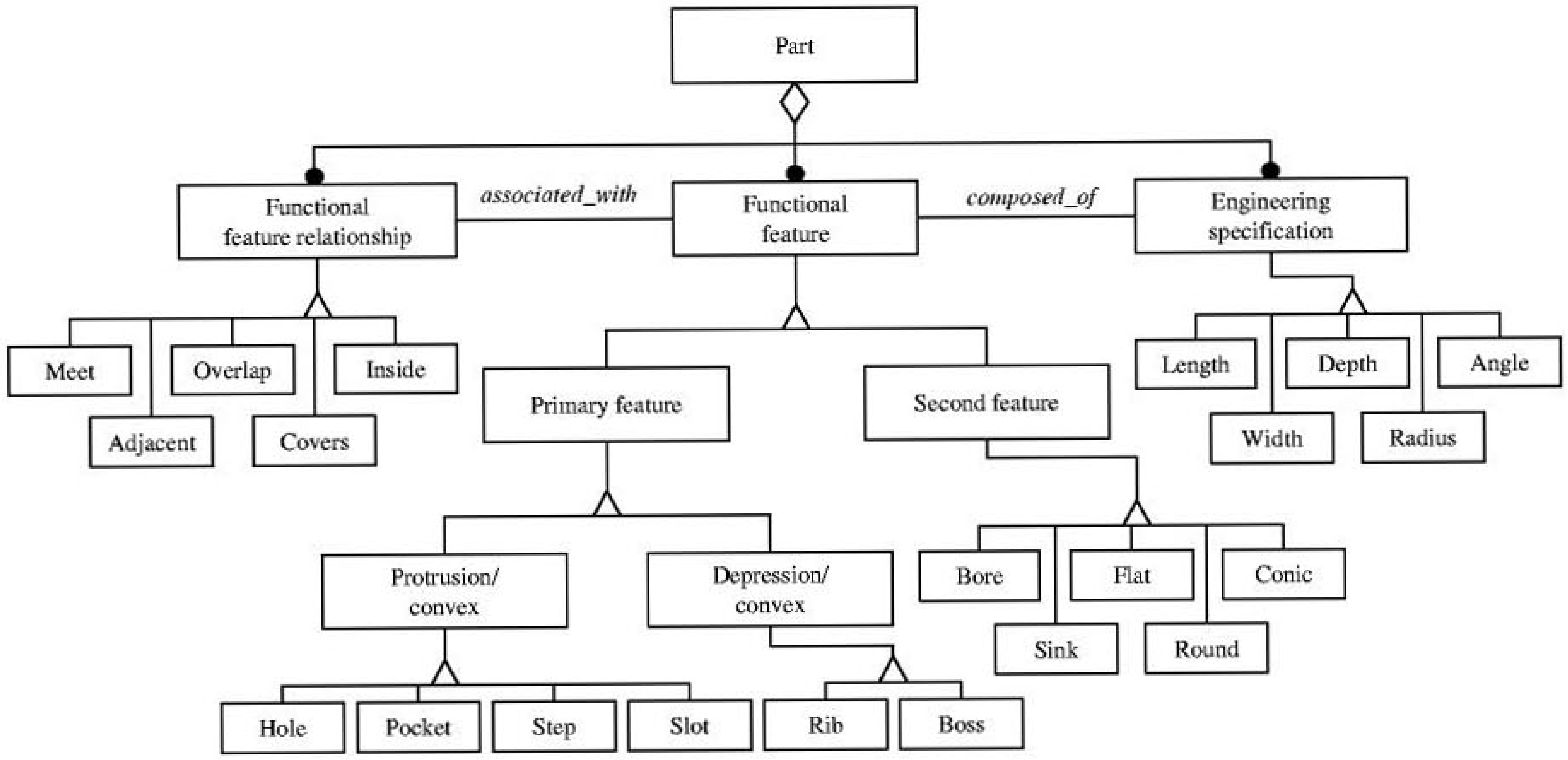


Figure 4. Structure of a part model.

(a)

Primary feature						Secondary feature				
Hole	Pocket	Step	Slot	Rib	Boss	Bore	Sink	Flat	Round	Conic
1	0	0	1	0	0	0	0	0	0	0

(b)

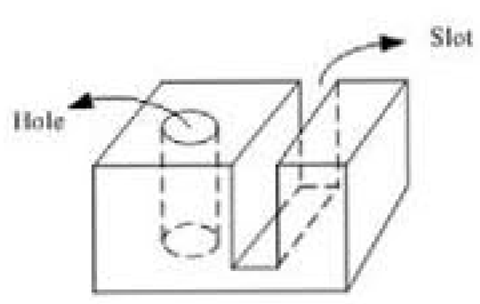


Figure 5. (a) Example of binary code-based representation for functional features and (b) a sample part.

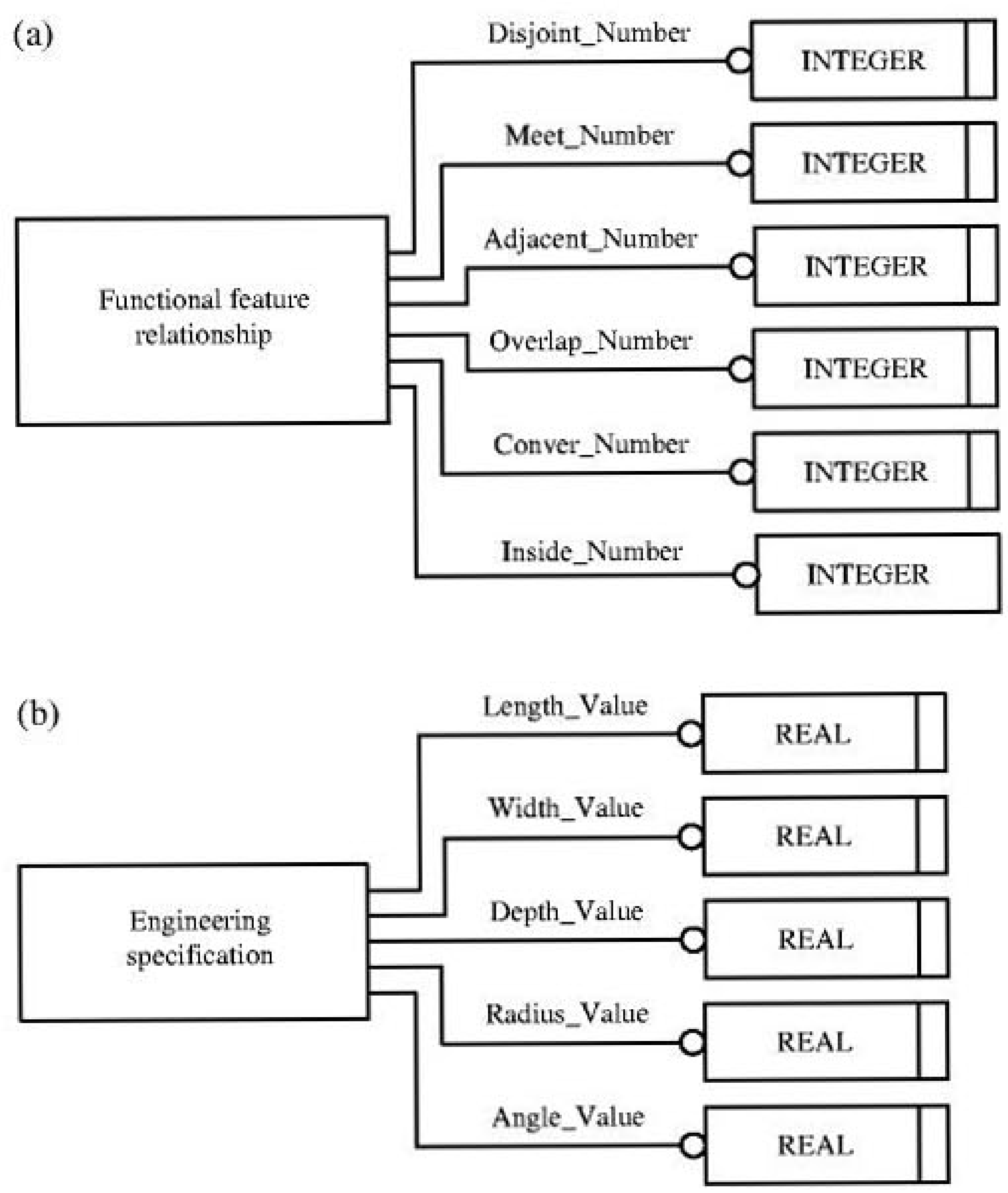


Figure 6. (a) EXPRESS for feature relationship (relationship number) and (b) EXPRESS for engineering specification (specification value).

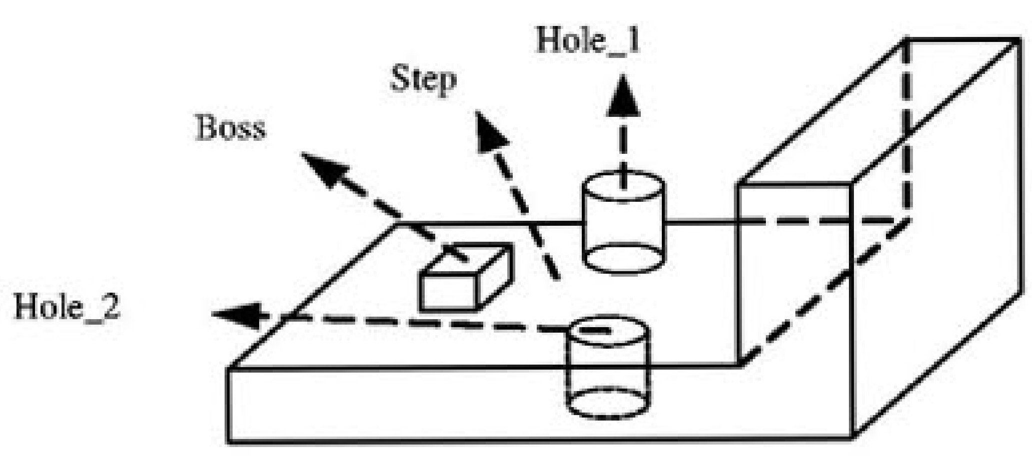
with two functional features, and every functional feature is composed of several engineering specifications. The feature relationship class is specialized into subclasses of 'meet,' 'adjacent,' 'overlap,' 'covers,' and 'inside' based on the spatial relationships between the functional features. Moreover, the functional feature can be divided into two types: (i) primary feature and (ii) secondary feature. The first type includes the functional features 'hole,' 'pocket,' 'step,' and 'slot,' while the second type includes the functional features 'rib' and 'boss.' For engineering specification, it is roughly classified into 'length,' 'width,' 'depth,' 'radius,' and 'angle.'

A part is characterized using a list of functional features, which can be treated as binary variables. Based on the functional features defined above, these functional features are required to describe specific part features. When coding a part based on the list, 'one' means that the part has a given functional feature while 'zero' indicates that it does not. Figure 5(a) shows the binary code-based representation of the functional features of the sample part illustrated in Figure 5(b).

Besides functional feature representation, the EXPRESS language, which supports the descriptions of the feature relationship (i.e., relationship number) and the engineering specification, is defined in Figure 6(a) and (b). This EXPRESS represents each entity instance using a rectangle, while each data type

(a)

```
SCHEMA Functional Feature Relationship;
    ENTITY Functional Feature Relationship;
        Meet_Number : 0;
        Adjacent_Number : 2;
        Overlap_Number : 0;
        Conver_Number: 0;
        Inside_Number: 1;
    END_ENTITY;
END_SCHEMA
```



Relationship (Step, Hole_1) = Adjacent
Relationship (Step, Boss) = Adjacent
Relationship (Step, Hole_2) = Inside

(b)

```
SCHEMA Engineering Specification;
    ENTITY Functional Feature_Slot;
        Length_Value : 4.55;
        Width_Value : 2.25;
        Depth_Value : 3.80;
        Radius_Value: Null;
        Angle_Value : Null;
    END_ENTITY;
END_SCHEMA
```

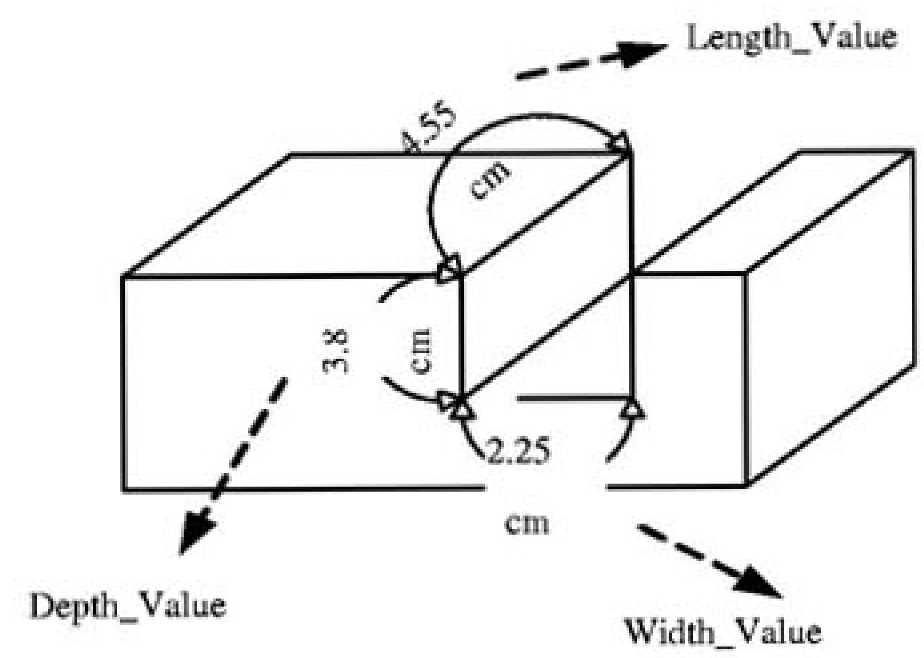


Figure 7. (a) Example of feature relationship representation in EXPRESS and (b) example of engineering specifications representation in EXPRESS.

is represented using a rectangle with a right double-line. According to the above EXPRESS, two examples with the structures of the feature relationship and the engineering specification in EXPRESS syntax are presented in Figure 7.

3.3 An Integrated Clustering Approach

This study combines the ART1 neural network, the GA, and the fuzzy ART techniques into an integrated clustering approach to effectively search out the most similar design case from the reference design case repository by querying functional features (including feature relationships) and engineering specifications. Their details are further discussed as follows.

3.3.1 ART1 NEURAL NETWORK FOR FUNCTIONAL FEATURE-BASED CLUSTERING OF SIMILAR DESIGN CASES

The ART1 neural network [2,11] is adopted as the first clustering technique for solving the problem of functional feature-based clustering of similar design cases. The ART1 neural network can be introduced in terms of ART1 characteristics, ART1 architecture, and ART1 algorithm, respectively. Moreover, an illustrative example involving the application of ART1 neural network to functional feature-based similar design case clustering is given.

ART1 CHARACTERISTICS

- Binary vector space representation: ART1 can process patterns expressed as vectors with components of either 0 or 1.
- Stability and plasticity: The ART1 network is sufficiently stable to preserve significant past learning, while remaining adaptable enough to incorporate new clusters whenever they appear.
- Unsupervised learning: In unsupervised learning, no environmental feedback exists to indicate the nature or correctness of network outputs. The network must discover for itself any significant relationships that may exist in the input data and translate the discovered relationships into outputs.
- Quick learning capability: In ART1 network, if the input pattern resembles any earlier exemplar, then it is clustered with that cluster. However, if the input pattern does not resemble any earlier exemplar, then a new cluster is created to represent that pattern.

ART1 ARCHITECTURE

Figure 8 shows the architecture of ART1. The network consists of two layers: the comparison layer and the recognition layer. The input values to the ART1 network are binary values. Moreover, each unit in the classification layer corresponds to a cluster, and units in the two layers are completely connected. Two types of connections exist: bottom up and top down. Units in the recognition layer also are connected by lateral connections. The ART1 architecture has three additional modules: Gain1, Gain2, and Reset, that provide control functions needed for training and classification. The comparison layer receives the binary vector, and the recognition layer compares and classifies the pattern.

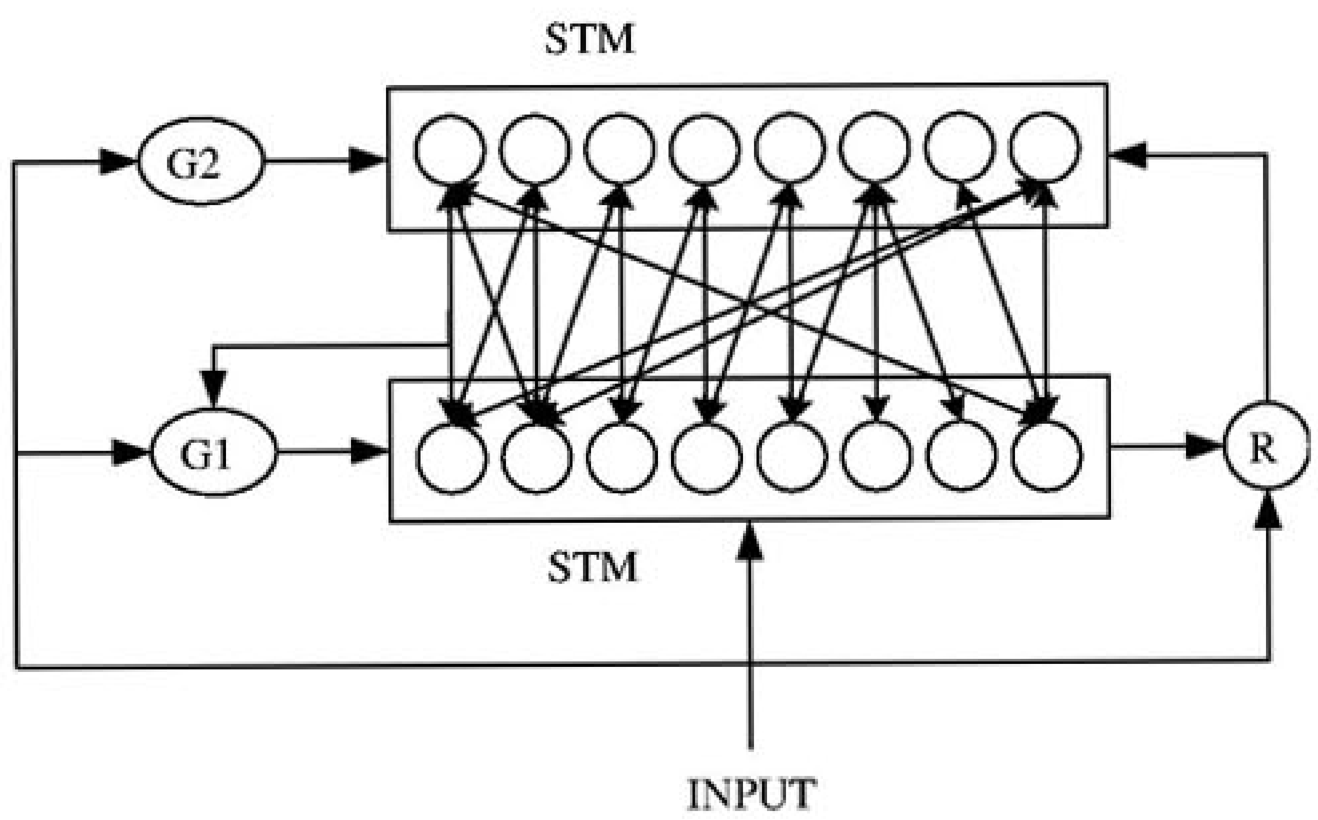


Figure 8. Adaptive resonance theory (ART1) architecture.

ART1 ALGORITHM

The ART 1 algorithm is applied to discover clusters of a set of binary-based pattern vectors. The algorithm can be summarized via the following steps:

Step 1. Initialize parameters:

$$L > 1, \quad t_{ij}(0) = 1, \quad b_{ij}(0) = \frac{L}{L - 1 + N}$$
$$\text{where } 0 \leq i \leq N - 1 \text{ and } 0 \leq j \leq M - 1 \text{ set } 0 \leq \rho \leq 1 \tag{1}$$

In Equation (1), $b_{ij}(t)$ denotes the bottom-up and $t_{ij}(t)$ the top-down connection weight between input node i and output node j at time t. Meanwhile, the fraction ρ represents the vigilance threshold, which indicates how close the input vector must be to a stored exemplar being considered to match it.

Step 2. Feed a new input.

Step 3. Compute the output, which is given by

$$\mu_j = \sum_{i}^{N-1} b_{ij}(t)x_j, \quad 0 \leq j \leq M - 1 \tag{2}$$

where μ_j denotes the output of output node j and x_i represents element i of the input vector.

Step 4. Select the best matching exemplar:

$$\mu_j^* = \max(\mu_j) \tag{3}$$

Step 5. Use the following vigilance test:

$$||x|| = \sum_{i=0}^{N-1} x_i \tag{4}$$

$$||T \cdot x|| = \sum_{i=0}^{N-1} t_{ij}x_j \tag{5}$$

if similarity $\nu = (||T \cdot x||/||x||) > \rho$ then proceed to Step 7, otherwise go to Step 6.

Step 6. Disable the best matching exemplar. The output of the best matching node selected in Step 4 is temporarily set to zero, and no longer participates in the maximization performed in Step 4. Then go to Step 3.

Step 7. Adapt the best matching exemplar:

$$t_{ij}(k+1) = t_{ij}^*(k)x_i, \quad b_{ij}(k+1) = \frac{t_{ij}(k)x_j}{0.5 + \sum_{i=0}^{N-1} t_{ij}^*(k)x_i} \tag{6}$$

Step 8. Enable any node disabled in Step 6. Repeat by returning to Step 2.

AN ILLUSTRATIVE EXAMPLE

The functional features of ten design cases listed in Table 1 are chosen as an example illustrating the ART1 algorithm for functional feature-based similar design case clustering, as discussed below.

From the initialization process of ART1, the initial weights are $t_{i0} = \{1, 1, 1, 1, 1, 1, 1, 1, 1, 1\}$ and $b_{i0} = 1/12\{1, 1, 1, 1, 1, 1, 1, 1, 1, 1\}$. Meanwhile, the vigilance parameter ρ is set to 0.5. The ten input samples are then fed to the ART1 algorithm individually. At least ten free output nodes are assumed to be available.

Sample 1 (Case 01): when sample 1 is fed, the one among the ten output nodes with the largest output is denoted as number 1. Since matching value $\mu_1^* = 5/12 = 0.417$ and similarity $\nu = 5/5 = 1.00 > \rho = 0.5$, the sample 1 is assigned to the first cluster. Additionally, the weights are then changed based on Equation (1), as $t_{i1} = \{1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0\}$ and $b_{i1} = 1/5.5\{1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0\}$.

Sample 2 (Case 02): when sample 2 is fed, no top layer node is competing clustering since only one active node exists; that is, node 1 is the unconditional winner. The matching value $\mu_1^* = 2/5.5 = 0.364$ and the similarity $\nu = 2/5 = 0.40 < \rho = 0.5$; therefore it fails the test, and sample 2 is considered a new cluster represented by another output node, number 2. The corresponding weights $t_{i2} = \{0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0\}$ and $b_{i2} = 1/4.5\{0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0\}$.

After all the samples in the Table 1 are fed in order, their matching value μ_j^*, similarity ν, top-down weight t_{ij}, and bottom-up weight b_{ij} are calculated and presented in table form, as shown in Table 2.

Table 1. Samples of binary code-based functional features of design cases.

	Primary Feature						Secondary Feature				
Case No	**Boss**	**Rib**	**Hole**	**Pocket**	**Step**	**Slot**	**Bore**	**Conic**	**Flat**	**Round**	**Sink**
01	1	0	1	0	0	1	0	0	1	1	0
02	0	1	1	1	0	1	0	1	0	0	0
03	0	0	1	1	0	1	1	0	0	0	0
04	0	1	1	0	1	1	0	0	0	1	0
05	1	0	0	0	1	0	1	0	0	0	0
06	0	0	0	0	1	0	1	1	0	0	0
07	1	0	0	0	1	0	0	1	0	1	0
08	1	0	0	0	0	0	0	1	0	1	0
09	1	0	0	0	1	0	0	1	0	0	0
10	1	0	1	0	1	0	0	0	0	0	0

Table 2. Clustering result of ten functional feature-based design cases.

Case	μ_j^*			$v=\frac{\lVert T\cdot x\rVert}{\lVert x\rVert}$			t_{ij}	b_{ij}	**Cluster**
01	$\mu_1^*=5/12$			$v_1^*=5/5>\rho$			$t_{i1}=\{1,0,1,0,0,1,0,0,1,1,0\}$	$b_{i1}=1/5.5\{1,0,1,0,0,1,0,0,1,1,0\}$	1
02	$\mu_1^*=2/5.5$			$v_1^*=2/5<\rho$			$t_{i2}=\{0,1,1,1,0,1,0,1,0,0,0\}$	$b_{i2}=1/5.5\{0,1,1,1,0,1,0,1,0,0,0\}$	2
03	$\mu_1=2/5.5$	$\mu_2^*=3/4.5$		$v_1=2/4$	$v_2^*=3/4>\rho$		$t_{i2}=\{0,0,1,1,0,1,1,0,0,0,0\}$	$b_{i2}=1/5.5\{0,0,1,1,0,1,1,0,0,0,0\}$	2
04	$\mu_1^*=3/5.5$	$\mu_2=2/4.5$		$v_1^*=3/5>\rho$	$v_2=2/5$		$t_{i1}=\{0,1,1,0,1,1,0,0,0,1,0\}$	$b_{i1}=1/5.5\{0,1,1,0,1,1,0,0,0,1,0\}$	1
05	$\mu_1=1/5.5$	$\mu_2^*=1/4.5$		$v_1=1/3$	$v_2^*=1/3<\rho$		$t_{i3}=\{1,0,0,0,1,0,1,0,0,0,0\}$	$b_{i3}=1/3.5\{1,0,0,0,1,0,1,0,0,0,0\}$	3
06	$\mu_1=1/5.5$	$\mu_2=1/4.5$	$\mu_3^*=1/3.5$	$v_1=1/3$	$v_2=1/3$	$v_3^*=2/3>\rho$	$t_{i3}=\{0,0,0,0,1,0,1,1,0,0,0\}$	$b_{i3}=1/3.5\{0,0,0,0,1,0,1,1,0,0,0\}$	3
07	$\mu_1=2/5.5$	$\mu_2=0/4.5$	$\mu_3^*=2/3.5$	$v_1=2/4$	$v_2=0/4$	$v_3^*=2/4>\rho$	$t_{i3}=\{1,0,0,0,1,0,0,1,0,1,0\}$	$b_{i3}=1/4.5\{1,0,0,0,1,0,0,1,0,1,0\}$	3
08	$\mu_1=1/5.5$	$\mu_2=0/4.5$	$\mu_3^*=3/4.5$	$v_1=1/3$	$v_2=0/3$	$v_3^*=3/3>\rho$	$t_{i3}=\{1,0,0,0,0,0,0,1,0,1,0\}$	$b_{i3}=1/3.5\{1,0,0,0,0,0,0,1,0,1,0\}$	3
09	$\mu_1=1/5.5$	$\mu_2=0/4.5$	$\mu_3^*=2/3.5$	$v_1=1/3$	$v_2=0/3$	$v_3^*=2/3>\rho$	$t_{i3}=\{1,0,0,0,1,0,0,1,0,0,0\}$	$b_{i3}=1/3.5\{1,0,0,0,1,0,0,1,0,0,0\}$	3
10	$\mu_1=2/5.5$	$\mu_2=1/4.5$	$\mu_3^*=2/3.5$	$v_1=2/3$	$v_2=1/3$	$v_3^*=2/3>\rho$	$t_{i3}=\{1,0,1,0,1,0,0,0,0,0,0\}$	$b_{i3}=1/3.5\{1,0,1,0,1,0,0,0,0,0,0\}$	3

The clustering result in Table 2 shows that these design cases are grouped into three categories: the first category contains Case01 and Case04, the second category includes Case02 and Case03, and the third category comprises Case05, Case06, Case07, Case08, Case09, and Case10. In this illustrative example, if a query pattern is $\{1,0,1,0,1,0,0,0,1,0,0\}$, then Case05, Case06, Case07, Case08, Case09, and Case10 are similar design cases to this query pattern in functional features.

3.3.2 GENETIC ALGORITHM FOR FEATURE RELATIONSHIP-BASED CLUSTERING OF SIMILAR DESIGN CASES

The GA has also been used in developing clustering techniques with specific domains [6,10]. This study applies the genetic clustering algorithm to deal with the problem of feature relationship-based similar design case clustering due to the characteristic of processing data for feature relationship in clustering presented in the following paradigm. The key steps of the GA should be summarized before illustrating the example.

CONCEPT OF GENETIC ALGORITHM

The GA belongs to a class of search techniques that mimic the principles of natural selection to develop solutions to large optimization problems. The genetic algorithm operates by maintaining and manipulating a population of potential solutions known as chromosomes. Each chromosome has an associated fitness value, which is a qualitative measure of the goodness of the associated solution. The fitness value is used to guide the stochastic selection of chromosomes, which are then used to generate new candidate solutions through crossover and mutation. Crossover generates new chromosomes by combining the selection of two or more selected parents. Mutation acts by randomly selecting genes, which are then altered, thereby preventing sub-optimal solutions from persisting and increasing population diversity. The process of selection, crossover, and mutation continues for a fixed number of generations or until a termination condition is satisfied.

GENETIC CLUSTERING ALGORITHM

The genetic clustering algorithm involves two stages. The first stage is the nearest neighbor algorithm. Objects are grouped in the nearest neighbor algorithm based on the average of the nearest neighbor distances. The nearest neighbor algorithm is used during the first stage to reduce the computational time and space. The second stage consists of a heuristic method and a genetic algorithm. Meanwhile, the heuristic method is used to identify a good clustering by applying the GA. The GA contains an initialization step and the iterative generations with three phases, namely the reproduction, crossover, and mutation phases. These phases are described as follows.

Initialization Step: A set of chromosomes is randomly generated in the initialization step. This set of chromosomes is termed the population. Each chromosome contains m bits, where the value of m is the number of components in the data set.

Reproduction Step: The fitness of each chromosome is calculated during this phase. After calculating the fitness for each chromosome in the population, the reproduction operator is implemented using a roulette wheel with slots sized based on fitness. The roulette wheel selection can be visualized by imagining a wheel on which each chromosome occupies an area related to its fitness value. When the wheel stops spinning, a fixed marker determines which chromosome will be selected to reproduce into the mating pool. Some critical formulas in this step are defined as:

$$T = \{B_i | c_i = 1, 1 \leq i \leq u\} \tag{7}$$

$$T' = \{B'_j | c'_j = 0, 1 \leq j \leq v\} \tag{8}$$

where T denotes one set of including 1's element in the chromosome, while T' represents other set of including 0's element in the chromosome.

$$\mathrm{SCORE}(C_i) = D_{\mathrm{int}\,er}(C_i)w - D_{\mathrm{int}\,ra}(C_i) \tag{9}$$

where $D_{\mathrm{int}\,er}(C_i)$ represents the maximum distance between two elements in the cluster C_i, and $D_{\mathrm{int}\,er}(C_i)$ represents the minimum distance between cluster C_i and the other clusters.

$$\mathrm{Fitness}(R) = \frac{\sum_{i=1}^{u} \mathrm{SCORE}(C_i)}{u} \tag{10}$$

Crossover Phase: Chromosomes are chosen through pairs. For each chosen pair, two random numbers are generated to decide which pieces of the chromosomes are to be interchanged. Assuming that the length of the chromosome is m, each random number is an integer in [1, m]. If a pair of chromosomes R and Q are chosen for applying the crossover operator, two random numbers e and f in [1, m] are generated to decide which pieces of the chromosomes require interchanging. Suppose $e<f$, then the bits from positions e to f of chromosome R are interchanged with those bits in the same positions of chromosome Q.

Mutation Phase: During the mutation phase, the bits of the chromosomes in the population are chosen from [1, m] with probability P_m. Each chosen bit is then changed from 0 to 1 or from 1 to 0. If the chosen bit is b_i, then for $1 \leq i \leq m$, it indicates a chosen cluster being discarded or produced in a chromosome.

AN ILLUSTRATIVE EXAMPLE

Based on the result of processing ART1 with the above example, the similar design cases (i.e., Cases 05, 06, 07, 08, 09, and 10) can be further refined through the genetic clustering algorithm. Table 3 presents the feature relationships between functional features design cases possess. Moreover, the feature relationship as a query pattern is also defined at the bottom of Table 3.

The numbers of different feature relationships for each design case are counted using the feature relationships between features in Table 3, as listed in Table 4.

At first, an initial population of size 4 is randomly selected as shown below, and the mutation probability p_m is set as 0.005.

Chromosome	Case 05	Case 06	Case 07	Case 08	Case 09	Case 10	Query
R_1	0	1	0	0	1	0	0
R_2	1	0	0	0	0	0	1
R_3	1	0	1	1	0	0	0
R_4	0	0	0	0	1	1	0

Following calculating the fitness of each individual chromosome based on Equation (10), Table 5 lists the results.

Chromosome numbers R_1, R_2, and R_4 have the highest fitness values. Deleting the one with the least fitness value (i.e., Chromosome R_3) provides a temporary reduced population ready to undergo reproduction and crossover. Pairs of chromosomes are now chosen at random: R_1 is paired with R_2, R_3 with R_4. A crossover section for each pair of chromosomes is randomly selected and marked by a '↑'. Table 6 lists the result of the processing crossover.

For the mutation process, the random value of mutation probability p_m is assumed to be 0.001, which is below the initial setting value $p_m = 0.005$. Thus, no bits undergo mutation at this probability value 0.001 during this run of GA.

Re-running the algorithm from the same reproduction phase, the fitness values for four new populations in Table 6 are determined, as displayed in Table 7.

Due to the R_1's least fitness value, the chromosome R_1 is deleted and replaced by R_3, which has the highest fitness value, 0.0089. Subsequently, performing crossover from the fourth to the seventh bits in two random pairs of chromosomes $\{R_1, R_2\}$ and $\{R_3, R_4\}$, producing another four new chromosomes, as shown in Table 8.

In this time of running GA, the random mutation probability p_m is assumed to be higher than the setting value $p_m = 0.005$, and the chromosome R_3 is mutated in its third bit (i.e., Case07). In this example, illustrated in Table 9, the third bit value '0' of chromosome R_3 thus is replaced by the value '1'.

Table 3. Feature relationship between features.

				Feature Relationship				
Case_No	**Serial_No**	**Feature_No**	**Feature_Name**	**Meet**	**Adjacent**	**Overlap**	**Covers**	**Inside**
05	01	01	Boss	02				
05	02	01	Boss	01 05				
05	03	05	Step			04		
05	04	05	Step			03		
05	05	07	Bore	01				
06	01	05	Step		02			
06	02	05	Step		01		03	
06	03	05	Step				02 04	
06	04	07	Bore				03	05
06	05	08	Conic					04
07	01	01	Boss					03
07	02	01	Boss		03			
07	03	05	Step		02			01
07	04	08	Conic					
07	05	10	Round					
08	01	01	Boss				04	02 03
08	02	01	Boss					01 03
08	03	08	Conic					01 02
08	04	10	Round		05		01	
08	05	10	Round		04			
09	01	01	Boss			02		
09	02	01	Boss	03		01		
09	03	05	Step					
09	04	05	Step	01				
09	05	08	Conic					
10	01	01	Boss		02		05	
10	02	01	Boss		01 05			
10	03	03	Hole					
10	04	03	Hole					05
10	05	05	Step		02		01	04
Query	01	01	Boss		04			04
Query	02	01	Boss				04	03
Query	03	03	Hole					02
Query	04	05	Step		01		02	01
Query	05	09	Flat					

Table 4. Numbers of different feature relationships.

Case_No	Meet_Num	Adjacent_Num	Overlap_Num	Covers_Num	Inside_Num
05	2	0	1	0	0
06	0	1	0	2	1
07	0	1	0	0	1
08	0	1	0	1	3
09	1	0	1	0	0
10	0	2	0	1	1
Query	0	1	0	1	2

Table 5. Fitness value for each chromosome.

Chromosome	Case05	Case06	Case07	Case08	Case09	Case10	Query	Fitness
R_1	0	1	0	0	1	0	0	0.6447
R_2	1	0	0	0	0	0	1	1.4462
R_3	1	0	1	1	0	0	0	0.5648
R_4	0	0	0	0	1	1	0	0.6447

Table 6. Result after crossover phase.

Chromosome	Case05	Case06	Case07	Case08	Case09	Case10	Query
R_1	0	1	0	0	0	0	1
R_2	1	0	0	0	1	0	0
R_3	1	0	0	0	1	1	0
R_4	0	0	0	0	0	0	1
				↑ e < - - -	- - -	- - -	↑ - - - > f

Table 7. Fitness value for each chromosome.

Chromosome	Case05	Case06	Case07	Case08	Case09	Case10	Query	Fitness
R_1	0	1	0	0	0	0	1	−1.5067
R_2	1	0	0	0	1	0	0	−1.4028
R_3	1	0	0	0	1	1	0	0.0089
R_4	0	0	0	0	0	0	1	0.0000

Table 8. Result after crossover phase.

Chromosome	Case05	Case06	Case07	Case08	Case09	Case10	Query
R_1	1	0	0	0	1	0	0
R_2	1	0	0	0	1	1	0
R_3	1	0	0	0	0	0	1
R_4	0	0	0	0	1	1	0
				↑ e < - - -	- - -	- - -	↑ - - - > f

Table 9. Result after mutation phase.

Chromosome	Case05	Case06	Case07	Case08	Case09	Case10	Query
R_1	1	0	0	0	1	0	0
R_2	1	0	0	0	1	1	0
R_3	1	0	1	0	0	0	1
R_4	0	0	0	0	1	1	0

Repeating the phases of reproduction, crossover, and mutation in order until a satisfactory solution is reached, or a specified number of generations is considered. Here, the chromosome R_4 of the maximum fitness value is assumed to be $\{1, 0, 0, 0, 0, 0, 1\}$ in the final of several generations. It represents that the best clustering number is two (i.e., C_1 and C_2), and the centers of C_1 and C_2 are located at the first bit (i.e., Case05) and the seventh bit (Query) of the chromosome R_4 with the maximum fitness, respectively. The other bits of chromosome R_4 with the maximum fitness (including the second bit (Case06), third bit (Case07), fourth bit (Case08), fifth bit (Case09), and sixth bit (Case10)) are individually examined to determine which cluster they belong to. From the clustering samples shown in Table 4, the calculation process is detailed as follows.

Case06:

$$D_{6,5} = \|\text{Case06} - \text{Case05}\|$$

$$= \sqrt{(0-2)^2 + (1-0)^2 + (0-1)^2 + (2-0)^2 + (1-0)^2}$$

$$= \sqrt{11}$$

$$D_{6,q} = \|\text{Case06} - \text{Query}\|$$

$$= \sqrt{(0-0)^2 + (1-1)^2 + (0-0)^2 + (2-1)^2 + (1-2)^2}$$

$$= \sqrt{2}$$

$\because D_{6,5} > D_{6,q}$ $\therefore$Case06 is classified into clustering $2(C_2)$.

Case07 :

$$D_{7,5} = \|\text{Case07} - \text{Case05}\|$$
$$= \sqrt{(0-2)^2+(1-0)^2+(0-1)^2+(0-0)^2+(1-0)^2}$$
$$= \sqrt{7}$$
$$D_{7,q} = \|\text{Case07} - \text{Query}\|$$
$$= \sqrt{(0-0)^2+(1-1)^2+(0-0)^2+(0-1)^2+(1-2)^2}$$
$$= \sqrt{2}$$

$\because D_{7,5} > D_{7,q}$ $\therefore$Case07 is classified into clustering $2(C_2)$.

Case08 :

$$D_{8,5} = \|\text{Case08} - \text{Case05}\|$$
$$= \sqrt{(0-2)^2+(1-0)^2+(0-1)^2+(1-0)^2+(3-0)^2}$$
$$= \sqrt{16}$$
$$D_{8,q} = \|\text{Case08} - \text{Query}\|$$
$$= \sqrt{(0-0)^2+(1-1)^2+(0-0)^2+(1-1)^2+(3-2)^2}$$
$$= \sqrt{1}$$

$\because D_{8,5} > D_{8,q}$ $\therefore$Case08 is classified into clustering $2(C_2)$.

Case09 :

$$D_{9,5} = \|\text{Case09} - \text{Case05}\|$$
$$= \sqrt{(1-2)^2+(0-0)^2+(1-1)^2+(0-0)^2+(0-0)^2}$$
$$= \sqrt{1}$$
$$D_{9,q} = \|\text{Case09} - \text{Query}\|$$
$$= \sqrt{(1-0)^2+(0-1)^2+(1-0)^2+(0-1)^2+(0-2)^2}$$
$$= \sqrt{8}$$

$\because D_{9,5} < D_{9,q}$ $\therefore$Case09 is classified into clustering $1(C_1)$.

Case10 :

$$D_{10,5} = \|\text{Case10} - \text{Case05}\|$$
$$= \sqrt{(0-2)^2+(2-0)^2+(0-1)^2+(1-0)^2+(1-0)^2}$$
$$= \sqrt{11}$$
$$D_{10,q} = \|\text{Case10} - \text{Query}\|$$
$$= \sqrt{(0-0)^2+(2-1)^2+(0-0)^2+(1-1)^2+(1-2)^2}$$
$$= \sqrt{2}$$

$\because D_{10,5} > D_{10,q}$ $\therefore$Case10 is classified into clustering $2(C_2)$.

Conducting the above clustering simulation of the GA identifies two categories: one containing the samples Case05 and Case09, and the other containing the samples Query, Case06, Case07, Case08, and Case10. As mentioned in the above clustering result, Case06, Case07, Case08, and Case10 have a similar feature relationship to the Query Case.

3.3.3 FUZZY ART FOR ENGINEERING SPECIFICATION-BASED CLUSTERING OF SIMILAR DESIGN CASES

Unsupervised fuzzy ART [9,13] synthesizes fuzzy set theory and ART network, which can self-organize stable recognition categories in response to arbitrary sequences of analog input patterns. This study adopts the fuzzy ART algorithm for clustering engineering specification-based similar design cases. The following steps implement the algorithm:

Step 1. Initialize the parameters: weight vector w_{ij}, choice parameter α, fast learning β, and vigilance parameter ρ such that

$$w_{ij} = 1, \quad \alpha = 0.5, \quad \beta = 0.8, \quad \rho = 0.5 \tag{11}$$

Step 2. Feed a new input pattern S to the input nodes.

Step 3. For each input I and F_2 node j, calculating the choice function T_j defined as:

$$T_j(I) = \frac{\|I \wedge w_j\|}{\alpha + \|w_j\|} \tag{12}$$

where the fuzzy intersection $\wedge$ is

$$(p \wedge q)_i = \min(p_i, q_i) \tag{13}$$

Step 4. Make a category choice when at most one F_2 node can become active at a given time. The index J denotes the chosen category where

$$T_J = \max\{T_j : j = 1, \ldots, N\} \tag{14}$$

Step 5. Occur resonance and proceed to Step 7 if the subset-hood match function $|I \wedge w_j|/|I|$ of the chosen category meets the vigilance criterion:

$$\frac{||I \wedge w_j||}{||I||} \geq \rho \tag{15}$$

Otherwise, go to Step 6.

Step 6. Test whether other categories of the output layer exist, to provide similarity testing. If there are other categories for similarity testing, then go back to Step 4; otherwise yield a new category and set its weights.

The search process continues until the chosen J satisfies the matching criterion (Equation 15).

Step 7. Modify the weights as follows:

$$w_J^{\text{new}} = \beta(I \wedge w_J^{\text{old}}) + (1-\beta)w_J^{\text{old}} \tag{16}$$

which updates the weights of the Jth category.

To further elucidate how the fuzzy ART applies to the engineering specification-based similar design case clustering, an illustrative example is given as follows.

AN ILLUSTRATIVE EXAMPLE

According to the cases (Query Case, Case06, Case07, Case08, and Case10) obtained by using the GA technique to feature relationship-based similar design case clustering, these cases can be further refined in this phase, which is the engineering specification-based similar design case clustering. Table 10 shows the engineering specifications of five design cases. Due to the four functional features involved in the Query Case, the engineering specifications of these functional features 'Boss,' 'Hole,' 'Step,' and 'Flat' are considered only in this example during the engineering specification-based similar design case clustering.

Table 11 lists the engineering specifications of functional features – 'Boss,' its engineering specifications-based similar design case clustering with fuzzy ART clustering algorithm is performed, as follows.

Case07_01: when the first pattern is fed, its own engineering specifications are directly taken as weights: $w_{11}=3.45$, $w_{12}=6.25$, $w_{13}=2.65$.

Case07_02: when the second pattern is fed, the following output values are calculated based on Equations (12), (15), and (16):

$$T_1 = \frac{\|I \wedge w_j\|}{\alpha + \|w_j\|} = \frac{\|(5.25, 2.75, 3.25) \wedge (3.45, 6.25, 2.65)\|}{0.5 + \|(3.45, 6.25, 2.65)\|} = 0.6342$$

Since

$$\frac{\|I \wedge w_j\|}{\|I\|} = \frac{\|(5.25, 2.75, 3.25) \wedge (3.45, 6.25, 2.65)\|}{\|(5.25, 2.75, 3.25)\|} = 0.7614 > \rho = 0.5,$$

therefore, the weights must be modified as:

$$w_{11} = 0.8 * (5.25 \wedge 3.45) + 0.2 * 3.45 = 3.45$$

$$w_{12} = 0.8 * (2.75 \wedge 6.25) + 0.2 * 6.25 = 3.45$$

$$w_{13} = 0.8 * (3.25 \wedge 2.65) + 0.2 * 2.65 = 2.65$$

Table 10. Engineering specifications of design cases.

Case_No	Serial_No	Feature_No	Feature_Name	Length	Width	Depth	Radius	Angle
06	01	05	Step	15.25	7.45	5.30		
06	02	05	Step	5.15	3.25	11.50		
06	03	05	Step	4.45	2.00	8.75		
06	04	07	Bore	3.50	3.50	3.50		
06	05	08	Conic			2.50	3.45	30
07	01	01	Boss	3.45	6.25	2.65		
07	02	01	Boss	5.25	2.75	3.25		
07	03	05	Step	9.25	14.25	1.85		
07	04	08	Conic			3.00	2.55	45
07	05	10	Round			7.50	2.50	
08	01	01	Boss	12.25	6.85	3.65		
08	02	01	Boss	5.55	4.65	3.25		
08	03	08	Conic			4.15	2.35	
08	04	10	Round			4.25	2.55	
08	05	10	Round			13.15	3.00	
10	01	01	Boss	24.35	12.55	9.45		
10	02	01	Boss	7.55	4.95	3.55		
10	03	03	Hole			8.45	2.55	
10	04	03	Hole			15.55	9.25	
10	05	05	Step	4.65	3.15	3.45		
Query	01	01	Boss	2.55	5.55	5.75		
Query	02	03	Hole			5.50	2.50	
Query	03	05	Step	4.50	13.25	2.85		
Query	04	09	Flat	4.55		6.50		

Table 11. Engineering specifications of functional feature – boss.

Case_No	Serial_No	Feature_No	Feature_Name	Length	Width	Depth	Radius	Angle
07	01	01	Boss	3.45	6.25	2.65		
07	02	01	Boss	5.25	2.75	3.25		
08	01	01	Boss	12.25	6.85	3.65		
08	02	01	Boss	5.55	4.65	3.25		
10	01	01	Boss	24.35	12.55	9.45		
10	02	01	Boss	7.55	4.95	3.55		
Query	01	01	Boss	2.55	5.55	5.75		

Table 12. Clustering results of five engineering specification-based design cases.

Case _No	Serial _No	Feature _No	Feature _Name	Length	Width	Depth	Radius	Angle	Cluster
Query	01	01	Boss	2.55	5.55	5.75			1
07	01	01	Boss	3.45	6.25	2.65			1
07	02	01	Boss	5.25	2.75	3.25			1
08	02	01	Boss	5.55	4.65	3.25			1
10	02	01	Boss	7.55	4.95	3.55			1
Query	02	03	Hole			5.50	2.50		1
10	03	03	Hole			8.45	2.55		1
Query	03	05	Step	4.50	13.25	2.85			2
07	03	05	Step	9.25	14.25	1.85			2
Query	04	09	Flat	4.55		6.50			Null

Case08_01: when the third pattern is fed, the following output values are computed based on Equations (12), (15), and (16):

$$T_1 = \frac{\|I \wedge w_j\|}{\alpha + \|w_j\|} = \frac{\|(12.25, 6.85, 3.65) \wedge (3.45, 3.45, 2.65)\|}{0.5 + \|(3.45, 3.45, 2.65)\|} = 0.9174$$

Since

$$\frac{\|I \wedge w_j\|}{\|I\|} = \frac{\|(12.25, 6.85, 3.65) \wedge (3.45, 3.45, 2.65)\|}{\|(12.25, 6.85, 3.25)\|} = 0.3829 < \rho = 0.5,$$

therefore, the second neuron is generated and its weights are set to:

$$w_{21} = 12.25, \quad w_{22} = 6.85, \quad w_{23} = 3.65$$

After feeding all patterns for functional features – 'Boss,' 'Hole,' 'Step,' and 'Flat,' their clustering results are shown in the extreme right column of the Table 12, as follows.

Since the results in Table 12, Case07, Case08, and Case10 are similar to the Query Case in terms of the engineering specification of functional feature – 'Boss,' while Case10 is similar to the Query Case in terms of the engineering specification of functional feature – 'Hole.' Moreover, Case07 and Query Case have similar engineering specifications for the functional feature – 'Step.'

Subsequently, the intersection operator (∩) is used to identify the most similar design cases, namely Case07 and Case10.

3.4 Similar Design Cases Ranking

The vector space model [14] defines the similarity between two terms by the cosine of the angle between their two vectors. Therefore, this vector model is employed to calculate the degree of similarity between similar design cases acquired through functional feature and engineering specification-based similar design case clustering based on the query of functional features (including feature relationships) and engineering specifications. Meanwhile, the query vector $\boldsymbol{Q}$ and the similar case vector $\boldsymbol{C}$ for engineering specification of the functional features – 'Boss' and 'Step' can be defined as $\boldsymbol{Q} = (x_{1,Q}, x_{2,Q}, x_{3,Q})$ and $\boldsymbol{C}_j = (x_{1,j}, x_{2,j}, x_{3,j})$, while the query vector $\boldsymbol{Q}$ and the similar case vector $\boldsymbol{C}$ for engineering specification of functional feature – 'Hole' is represented by $\boldsymbol{Q} = (x_{1,Q}, x_{2,Q})$ and $\boldsymbol{C}_j = (x_{1,j}, x_{2,j})$.

Using the result discussed at the end of the Section 3.2.3 as an example, the correlation coefficient between Case07 and Query Case, as well as between Case10 and Query Case is calculated as:

$$\begin{aligned}\mathrm{Sim}_{\mathrm{Boss01}}(\boldsymbol{C}_{07}, \boldsymbol{Q}) &= \frac{\boldsymbol{C}_{07}\cdot \boldsymbol{Q}}{|\boldsymbol{C}_{07}| \times |\boldsymbol{Q}|}\\ &= \frac{(3.45, 6.25, 2.65)\cdot(2.55, 5.55, 5.75)}{\|(3.45, 6.25, 2.65)\| \times \|(2.55, 5.55, 5.75)\|}\\ &= 0.9193\end{aligned}$$

$$\begin{aligned}\mathrm{Sim}_{\mathrm{Boss02}}(\boldsymbol{C}_{07}, \boldsymbol{Q}) &= \frac{\boldsymbol{C}_{07}\cdot \boldsymbol{Q}}{|\boldsymbol{C}_{07}| \times |\boldsymbol{Q}|}\\ &= \frac{(5.25, 2.75, 3.25)\cdot(2.55, 5.55, 5.75)}{\|(5.25, 2.75, 3.25)\| \times \|(2.55, 5.55, 5.75)\|}\\ &= 0.8349\end{aligned}$$

Since $\mathrm{Sim}_{\mathrm{Boss01}}(\boldsymbol{C}_{07}, \boldsymbol{Q}) > \mathrm{Sim}_{\mathrm{Boss02}}(\boldsymbol{C}_{07}, \boldsymbol{Q})$, the $\mathrm{Sim}_{\mathrm{Boss01}}(\boldsymbol{C}_{07}, \boldsymbol{Q})$ is picked out to calculate the total similarity of Case07.

$$\begin{aligned}\mathrm{Sim}_{\mathrm{Step03}}(\boldsymbol{C}_{07}, \boldsymbol{Q}) &= \frac{\boldsymbol{C}_{07}\cdot \boldsymbol{Q}}{|\boldsymbol{C}_{07}| \times |\boldsymbol{Q}|}\\ &= \frac{(9.25, 14.25, 1.85)\cdot(4.50, 13.25, 2.85)}{\|(9.25, 14.25, 1.85)\| \times \|(4.50, 13.25, 2.85)\|}\\ &= 0.9658\end{aligned}$$

Therefore, the total similarity between Case07 and Query Case, $T\mathrm{Sim}(\boldsymbol{C}_{07}, \boldsymbol{Q})$ is $0.9193 + 0.9658 = 1.8851$.

$$\begin{aligned}\mathrm{Sim}_{\mathrm{Boss02}}(\boldsymbol{C}_{10}, \boldsymbol{Q}) &= \frac{\boldsymbol{C}_{10}\cdot \boldsymbol{Q}}{|\boldsymbol{C}_{10}| \times |\boldsymbol{Q}|}\\ &= \frac{(7.55, 4.95, 3.55)\cdot(2.55, 5.55, 5.75)}{\|(7.55, 4.95, 3.55)\| \times \|(2.55, 5.55, 5.75)\|}\\ &= 0.8250\end{aligned}$$

$$\begin{aligned}\mathrm{Sim}_{\mathrm{Hole03}}(\boldsymbol{C}_{10}, \boldsymbol{Q}) &= \frac{\boldsymbol{C}_{10}\cdot \boldsymbol{Q}}{|\boldsymbol{C}_{10}| \times |\boldsymbol{Q}|}\\ &= \frac{(8.45, 2.55)\cdot(5.55, 2.50)}{\|(8.45, 2.55)\| \times \|(5.55, 2.50)\|} = 0.9911\end{aligned}$$

Consequently, the total similarity between Case10 and Query Case, $T\mathrm{Sim}(\boldsymbol{C}_{10}, \boldsymbol{Q})$ is $0.8250 + 0.9911 = 1.8161$.

According to the above calculation results, Case07 resembles Query Case more than does Case10. Consequently, the degree of similarity to Query Case follows the order Case07 and Case10.

3.5 A Case-based Representation for Designed Entities

To effectively record and organize related product and information and engineering knowledge of a designed entity, and facilitate the engineering designers in accessing them easily and quickly, a object-oriented case representation for a designed entity is proposed. From Figure 9, a 'Case' is viewed as a box that contains related tags and links the product information and engineering knowledge of a design entity (i.e., an engineering model). The scheme of a case consists of three features: case feature, model feature, and semantic feature. Case feature defines the contents of case data, such as case name, ID, tag ID, name, model creator, contributor, date, language, version, and location. Meanwhile, model feature indicates the tag for product information that records the detailed information of a design entity, including customer requirements, functional requirements, and functional features. Finally, semantic feature represents the tags for engineering knowledge that also record the design knowledge and experience of engineering designers. These tags for engineering knowledge are classified into three categories, namely the tag for feature-based design, the tag for engineering change, and the tag for design by modification/reference. Each of these tags points to relevant production information or engineering knowledge.

4. Implementation and Experimental Example

Based on the integrated clustering approach for functional feature and engineering specification-based reference design retrieval, a prototype functional feature and engineering specification-based reference design retrieval mechanism was implemented at the Enterprise System Engineering Research Lab (ESERL) of National Cheng Kung University, Taiwan, ROC. The computer hardware used in the experiment comprised an Acer Veriton 7100 PC, while the software used to implement this mechanisms were (i) Microsoft Windows XP Professional as the development platform, (ii) Java Server Page (JSP) as the programming language, (iii) Dreamweaver MX 2004 as the programming development platform, (iv) Apache Tomcat 5.0 as the web server, and (v) Microsoft SQL Server 2000 for databases. Meanwhile, a SQL database served as the engineering design case base for storing related product information and engineering knowledge of a design case.

Figures 10, 11, and 12 present parts of the user interfaces of a functional feature and engineering specification-based reference design retrieval mechanism. Figure 10 shows the screen of functional feature and engineering specification description for the users. Figure 11 displays the screen of similarity ranking for retrieved cases; while Figure 12 illustrates the knowledge content of the most similar design case,

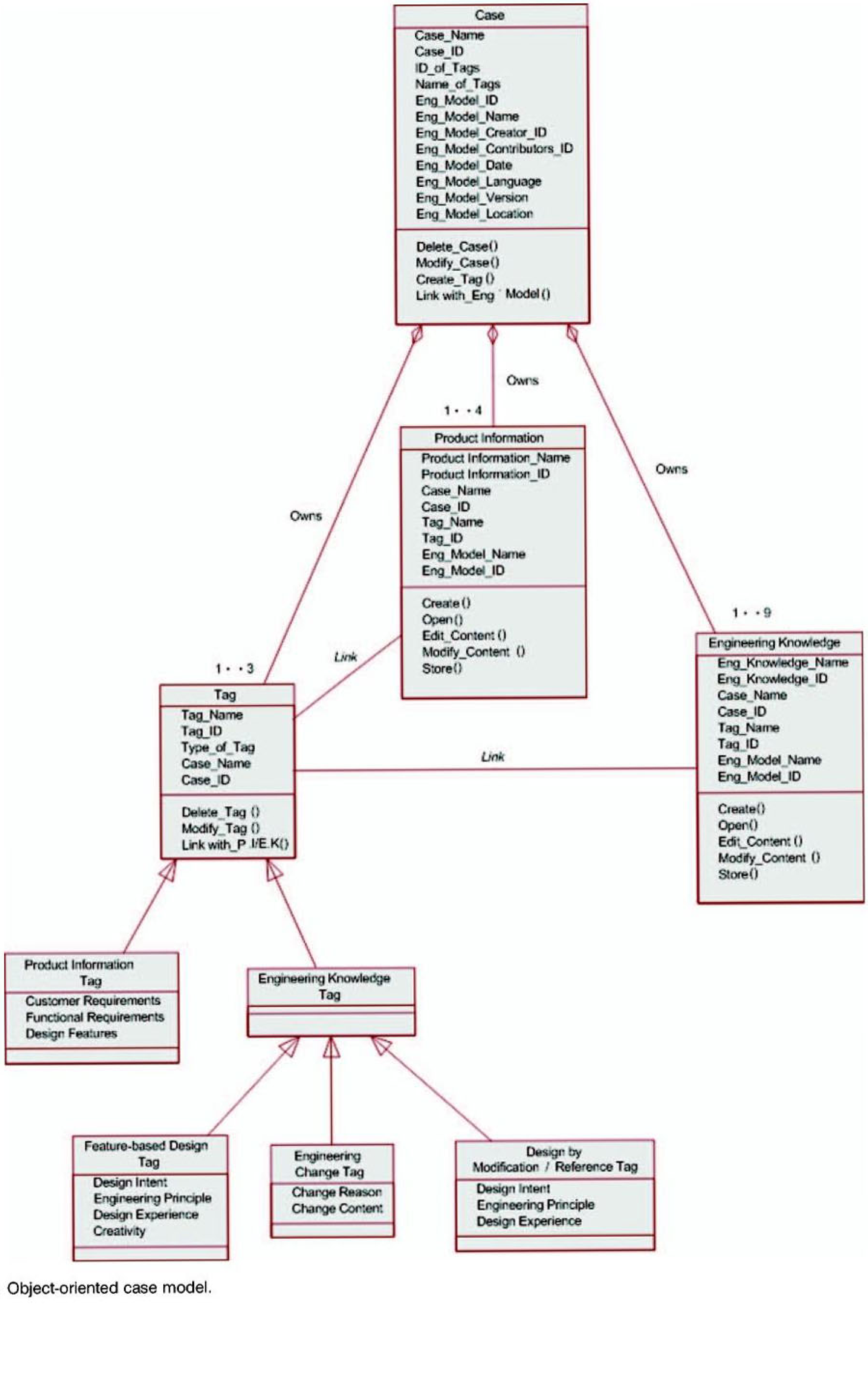


Figure 9. Object-oriented case model.

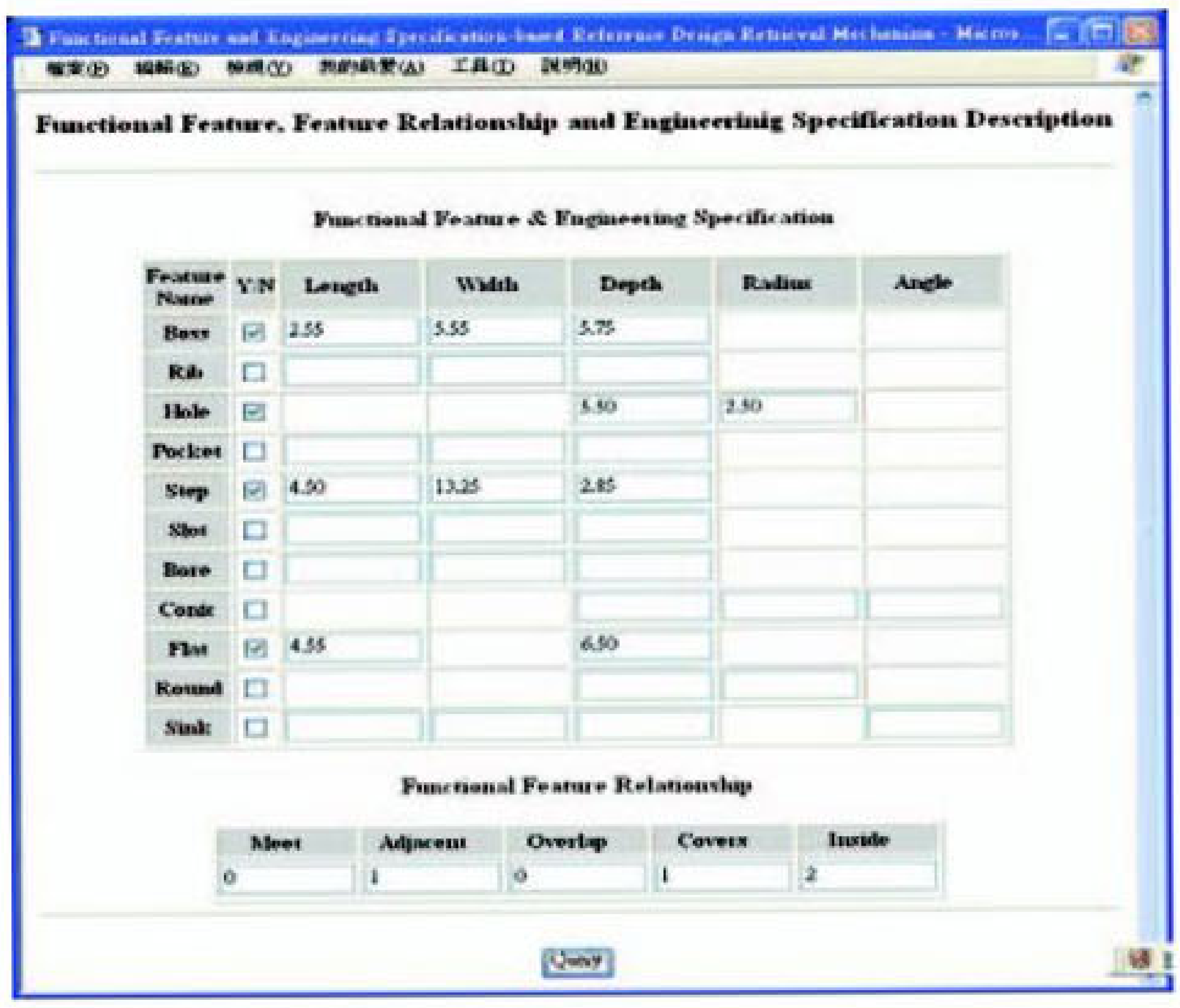


Figure 10. Functional feature and engineering specification description.

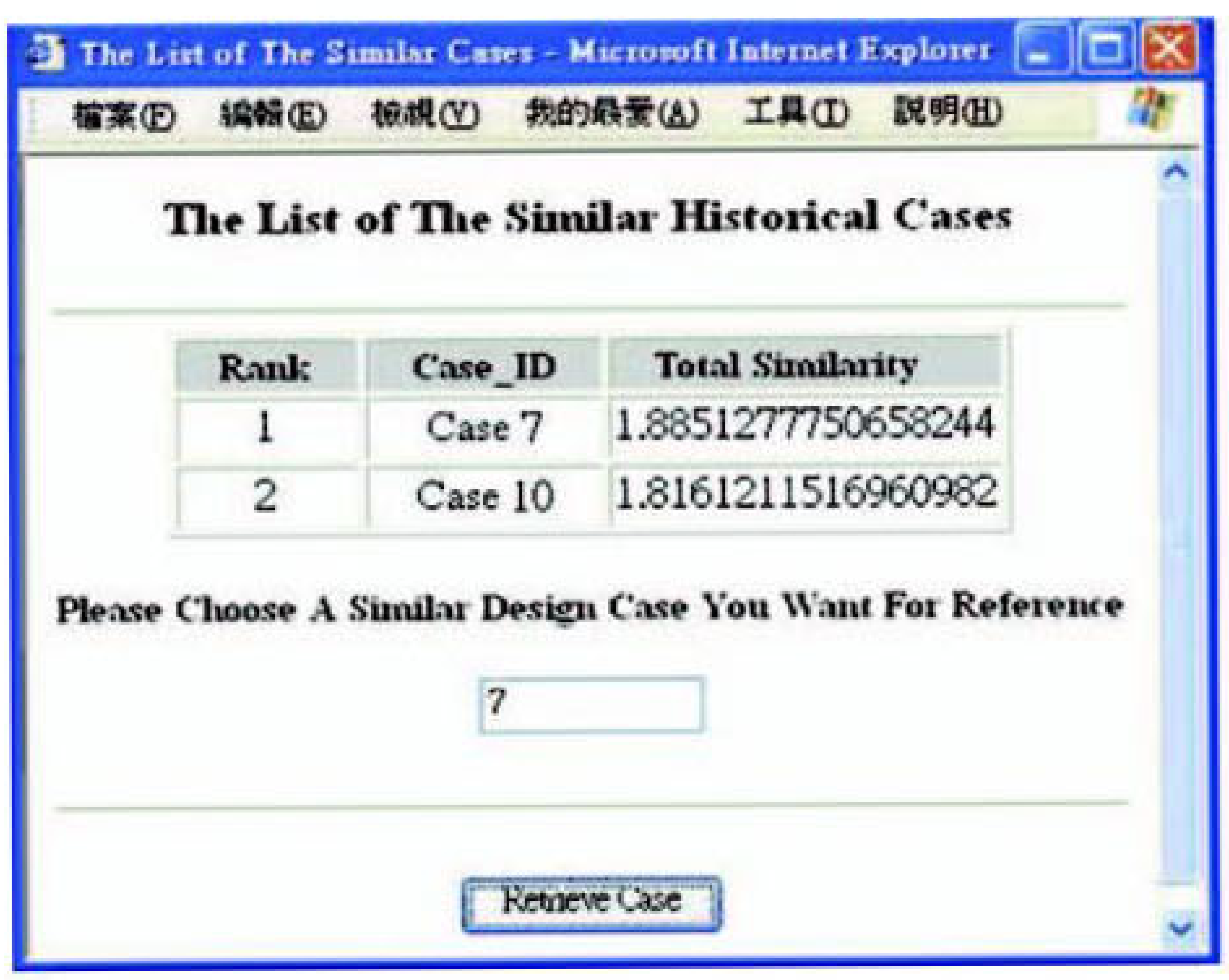


Figure 11. Similarity ranking for retrieved design cases.

including design intent and design experience of establishing functional features, feature relationships, and engineering specifications.

5. Concluding Remarks

This study first presents an engineering knowledge management framework, and then focuses on developing technology for functional feature and engineering specification-based reference design retrieval.

The tasks involved in the development include: (i) designing a functional feature and engineering design-based reference design retrieval process, (ii) developing a functional feature and engineering specification representation, (iii) investigating and integrating ART1 (adaptive resonance theory 1) neural network, GA (genetic algorithm), and fuzzy ART (fuzzy adaptive resonance theory) clustering techniques, and (iv) implementing a functional feature and engineering specification-based reference design retrieval mechanism and experimenting with an example.

The results in the study facilitate the sharing of engineering knowledge for engineering knowledge management purposes in engineering design environments. They can thus increase product development capability; reduce development cycle time and cost, and ultimately increase the product marketability.

Acknowledgment

The financial support through National Science Council of Taiwan (Grant No. 93-2212-E-006-021) is gratefully acknowledged.

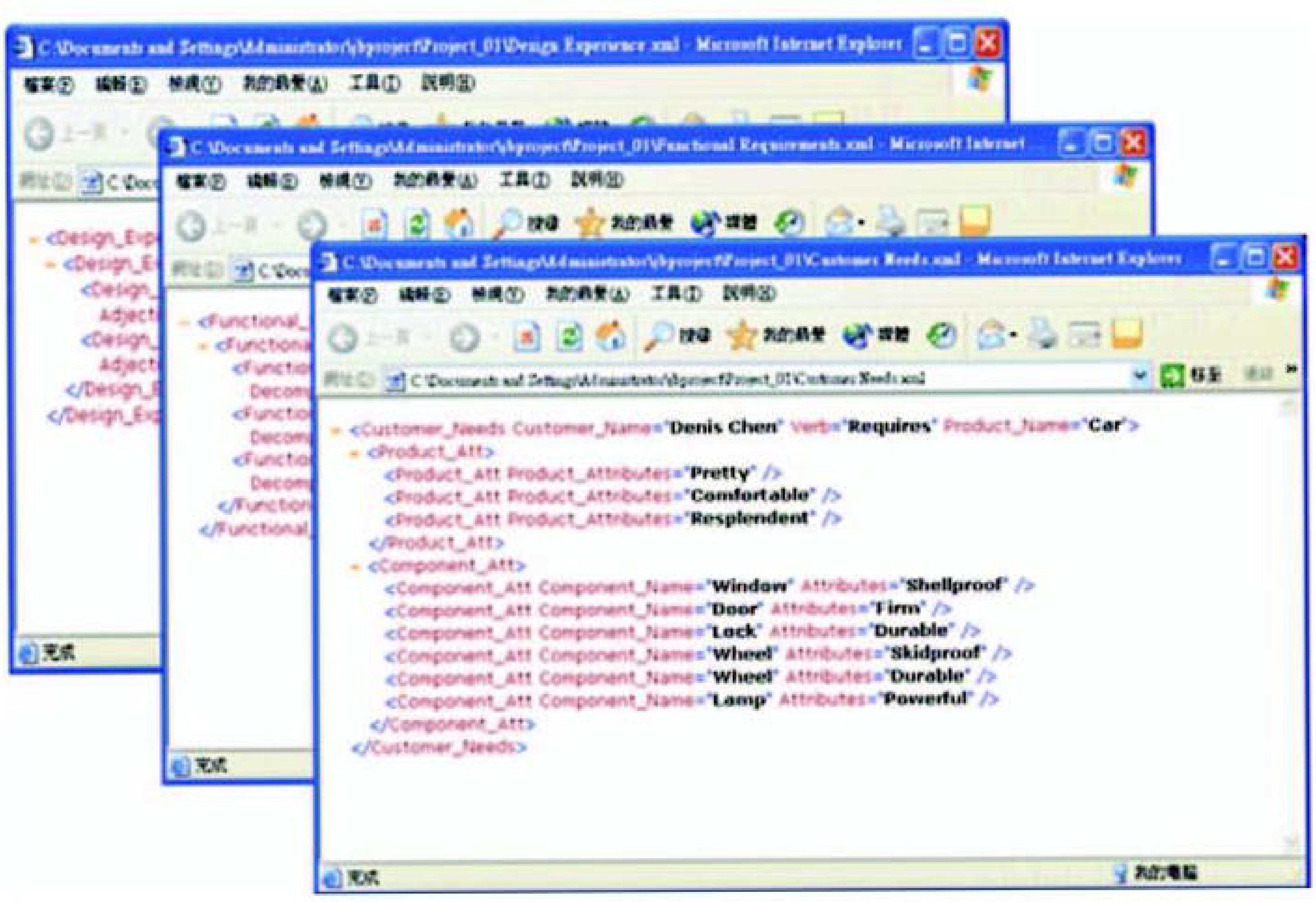


Figure 12. Related engineering knowledge of the most similar design case (Case07).

References

1. Booch, G., Rumbaugh, J. and Jacobson, I. (1999). The Unified Modeling Language User Guide, Reading, MA: Addition-Wesley.
2. Chang, C.A. and Tsai, C.Y. (1997). Using ART1 Neural Networks with Destructive Solid Geometry for Design Retrieving Systems, *Computers in Industry*, **34**(1): 27–41.
3. Chen, Y.M. and Hsiao, Y.T. (1997). A Collaborative Data Management Framework for Concurrent Product and Process Development, *International Journal Computer Integrated Manufacturing*, **10**(6): 446–469.
4. Dieter, G.E. (2000). *Engineering Design*, New York: McGraw-Hill.
5. Ertas, A. and Jones, J.C. (1996). The Engineering Design Process, 2nd edn, New York: Wiley.
6. Garai, G. and Chaudhuri, B.B. (2004). A Novel Genetic Algorithm for Automatic Clustering, *Pattern Recognition Letters*, **25**(2): 173–187.
7. Homer, G.R., Thompson, D.M. and Deacon, M. (2002). A Distributed Document Management System, *Computing & Control Engineering Journal*, **13**(6): 315–318.
8. Kirchmer, M. (1998). *Business Process Oriented Implementation of Standard Software*, New York: Springer-Verlag.
9. Kim, M.H., Jang, D.S. and Yang, Y.K. (2001). A Robust-invariant Pattern Recognition Model using Fuzzy ART, *Pattern Recognition*, **34**(8): 1685–1696.
10. Liu, Y., Chen, K., Liao, X. and Zhang, W. (2004). A Genetic Clustering Method for Intrusion Detection, *Pattern Recognition*, **37**(5): 927–942.
11. Massey, L. (2003). On the Quality of ART Text Clustering, *Neural Networks*, **16**(5–6): 771–778.
12. Prasad, B. (1997). Concurrent Engineering Fundamentals, Volume II: Integrated Product Development, USA: Prentice Hall.
13. Pacella, M., Semeraro, Q. and Anglanu, A. (2004). Manufacturing Quality Control by Means of a Fuzzy ART Network Trained on Natural Process Data, *Engineering Applications of Artificial Intelligence*, **17**(1): 83–96.
14. Ricardo, B.Y. and Berthier, R.N. (1999). *Modern Information Retrieval*, ACM Press.
15. Ullman, D.G. (2003). The Mechanical Design Process, 2nd edn, New York: McGraw-Hill.
16. Wang, C.B., Chen, Y.M., Chen, Y.J. and Ho, C.T. (2005). Methodology and System Framework for Knowledge Management in Allied Concurrent Engineering, *International Journal of Computer Integrated Manufacturing*, **18**(1): 53–72.
17. Wang, C.B., Chen, Y.J., Chen, Y.J. and Chu, H.C. (2005). Knowledge Refinement for Engineering Knowledge Management, *Concurrent Engineering: Research and Applications*, **13**(1): 43–56.
18. Wang, C.B., Chen, Y.J. and Chen, Y.J. (2004). On Technology for Customer Requirement-based Reference Design Retrieval, In: *The fifth Asia-Pacific Conference on Industrial Engineering and Management Systems.*
19. Wang, C.B., Chen, Y.J., Chen, Y.J. and Chu, H.C. (2005). Developing Technology for Functional Requirement-based Reference Design Retrieval in Engineering Design, In: *The 12th ISPE International Conference on Concurrent Engineering: Research and Applications.*

Dr Yuh-Jen Chen

Dr Yuh-Jen Chen is currently a postdoctoral fellow in the Institute of Manufacturing Engineering, College of Electrical Engineering and Computer Science, National Cheng Kung University, Taiwan, ROC. He received his PhD and MS degrees in Institute of Manufacturing Engineering of National Cheng Kung University in 2005 and 2001 respectively, and gained his BS degree from the Department of Mathematics of Chung Yuan Christian University, Taiwan, ROC, in 1999. His current research interests include Engineering Data and Knowledge Management Systems, Artificial Intelligent and Expert Systems, Supply Chain Simulation System, and e-Service.

Dr Yuh-Min Chen

Dr Yuh-Min Chen is currently a Professor and the Director of the Institute of Manufacturing Engineering, College of Electrical Engineering and Computer Science, National Cheng Kung University, Taiwan, ROC. He graduated from the Ohio State University with a PhD degree in Industrial and Systems Engineering in 1991 and received his MS and BS degrees from National Tsing Hua University, Taiwan, ROC, in 1981 and 1983 respectively. Before joining the faculty of Institute of Manufacturing Engineering in 1994, he worked as a research engineer in Structural Dynamics Research Corporation, USA for three years. His current research interests include Enterprise Integration, Engineering Data and Knowledge Management, Computer-Aided Concurrent Engineering, and Manufacturing Information Systems.

Dr Hui-Chuan Chu

Dr Hui-Chuan Chu is an Associate Professor of Department of Special Education, National University of Tainan in Taiwan, ROC. She received her PhD degree from Columbia University in 1998. Her research interests are Knowledge Management, Teacher Knowledge and Integration of Information Technology in Teacher Education.

Dr Ching-Bin Wang

Dr Ching-Bin Wang is currently an Associated Professor and the Chairman of Electronic Commerce Management Department, Nan Hua University in Taiwan, ROC. He received his PhD degree in Computer Science from the City University of New York in 1995, and gained his MS degree from University of Southern California and BS degrees from National Tsing Hua University, Taiwan, ROC, in 1985 and 1981 respectively. His research interests include Data Mining, Network Management, Engineering Data and Knowledge Management, and System Integration.

Dr Der-Chian Tsaih

Dr Der-Chian Tsaih is now associate professor in the Department of Electronic Commerce Management, Nan-Hua University, Chia-Yi, Taiwan. He received the BS degree in Electrical Engineering from the Chinese Culture University, Taipei, Taiwan, in 1984, the MS degree in Electrical Engineering from George Washington University, D. C., in 1989, and Ph.D degree in Electrical Engineering from Polytechnic University, Brooklyn, NY, in 1995. In 1996 he joined the faculty of Wu-Feng Technical Institute, as an associate professor in the Department of Information Management, Chia-Yi, Taiwan. His research interests include Wireless Network, Artificial Intelligence, and e-Learning.

Hui-Mei Yang

Hui-Mei Yang is currently an Instructor at the Department of Business Administration, Tatung Institute of Commerce and Technology, Chia-Yi, Taiwan. She graduated from the North Texas University with a MS degree in Economics. Now, her research interests include Economics, Knowledge Management, and e-Learning.